

# **SISTEM PAKAR DETEKSI DINI PENYAKIT BERBASIS WEB MENGUNAKAN METODE *VARIABLE-CENTERED* *INTELLIGENT RULE SYSTEM***

## **TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Teknik Pada  
Jurusan Teknik Informatika

Oleh :

**ARINA ARIFIN**  
**10451025501**



**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU  
PEKANBARU  
2011**

**SISTEM PAKAR UNTUK DETEKSI DINI PENYAKIT  
BERBASIS WEB MENGGUNAKAN METODE *VARIABLE  
CENTERED INTELLIGENT RULE SYSTEM***

**ARINA ARIFIN  
10451025501**

Tanggal Sidang : 15 Juni 2011  
Periode Wisuda : 14 Juli 2011

Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Sultan Syarif Kasim Riau

**ABSTRAK**

Sistem pakar adalah program pemberi nasehat yang terkomputerisasi yang ditujukan untuk meniru proses *reasoning* (pertimbangan) dan pengetahuan dari pakar dalam menyelesaikan permasalahan yang spesifik. Salah satu contoh penggunaan sistem pakar yaitu pada dunia kedokteran, yaitu sistem pakar untuk mendiagnosa suatu penyakit dan mencari solusi penyembuhannya.

Penelitian ini menggunakan sistem pakar yang dibangun menggunakan metode *Variable-Centered Intelligent Rule System* yang merupakan hasil perkawinan dari metode *Rule Base System* dan *Ripple Down Rule* dan menggunakan teknik *forward chaining* dalam proses inferensi. Sistem pakar ini menggunakan metode *Certainty Factor* untuk mengatasi ketidakpastian dalam pengambilan keputusan. Sistem ini dibangun berbasis web dengan menggunakan bahasa pemrograman PHP dan MySQL sebagai *databasenya*.

Hasil dari penelitian ini adalah sistem pakar dapat digunakan untuk deteksi dini penyakit berdasarkan gejala yang dirasakan serta memberikan solusi yang tepat. Berdasarkan pengujian, sistem sudah dapat mewakili seorang pakar dalam diagnosa awal penyakit.

Kata Kunci : *Certainty Factor*, Sistem Pakar, *Variable-Centered Intelligent Rule System*.

## DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN .....	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL .....	iv
LEMBAR PERNYATAAN .....	v
LEMBAR PERSEMBAHAN .....	vi
ABSTRAK .....	vii
ABSTRACT .....	viii
KATA PENGANTAR .....	ix
DAFTAR ISI .....	xi
DAFTAR GAMBAR .....	xvi
DAFTAR TABEL.....	xviii
DAFTAR RUMUS .....	xx
DAFTAR SINGKATAN .....	xxi
DAFTAR ISTILAH .....	xxii
DAFTAR LAMPIRAN.....	xiv
BAB I. PENDAHULUAN .....	I-1
1.1 Latar Belakang Masalah .....	I-1
1.2 Rumusan Masalah.....	I-2
1.3 Batasan Masalah .....	I-2
1.4 Tujuan Penelitian.....	I-3
1.5 Sistematika Pembahasan.....	I-3
BAB II. LANDASAN TEORI .....	II-1
2.1 Konsep Dasar Sistem.....	II-1
2.2 Sistem Pakar ( <i>Expert System</i> ).....	II-2
2.2.1 Konsep Dasar Sistem Pakar.....	II-2
2.2.2 Struktur Sistem Pakar .....	II-3
2.2.3 Komponen Sistem Pakar .....	II-3

2.3 <i>Variable-Centered Intelligent Rule System (VCIRS)</i> .....	II-5
2.3.1 Arsitektur Sistem .....	II-6
2.3.1.1 <i>Variable-Centered Rule Structure</i> .....	II-7
2.3.1.2 <i>Refinement Module</i> .....	II-8
2.3.2 Pembangunan Pengetahuan .....	II-10
2.3.2.1 Pohon Inferensi VCIRS .....	II-10
2.3.3 Faktor Kepastian ( <i>Certainty Factor</i> ) .....	II-12
2.3.4 Proses Inferensia .....	II-13
2.3.5 Evaluasi Sistem.....	II-13
2.4 Pemodelan Analisis .....	II-16
2.5 Penyakit.....	II-16
2.5.1 Penyakit Demam Berdarah Dengue (DBD) .....	II-17
2.5.1.1 Pengertian Penyakit DBD .....	II-17
2.5.1.2 Gejala Penyakit DBD .....	II-17
2.5.2 Penyakit Tipus .....	II-17
2.5.2.1 Pengertian Penyakit Tipus .....	II-18
2.5.2.2 Gejala Penyakit Tipus .....	II-18
2.5.3 Penyakit Malaria .....	II-18
2.5.3.1 Gejala Penyakit Malaria.....	II-18
2.5.4 Penyakit Maag .....	II-19
2.5.4.1 Gejala Penyakit Maag .....	II-19
BAB III. METODOLOGI PENELITIAN .....	III-1
3.1 Tahapan Penelitian .....	III-1
3.2 Pengumpulan Data.....	III-2
3.3 Analisa Kebutuhan Sistem.....	III-2
3.3.1 Analisa Masalah .....	III-3
3.3.2 Analisa Kebutuhan .....	III-3
3.3.3 Alat Bantu .....	III-3
3.4 Perancangan Perangkat Lunak.....	III-4
3.5 Implementasi .....	III-4
3.6 Pengujian .....	III-4

3.7 Kesimpulan dan Saran .....	III-4
BAB IV. ANALISA DAN PERANCANGAN .....	IV-1
4.1 Analisa Sistem Dengan Menggunakan VCIRS .....	IV-1
4.1.1 Analisa Masalah .....	IV-1
4.1.2 Analisa Sistem Baru .....	IV-2
4.1.3 Analisa Data Sistem .....	IV-2
4.1.4 Bagan Alir Sistem ( <i>Flowchart System</i> ) .....	IV-5
4.1.5 Struktur Basis Pengetahuan .....	IV-6
4.2 Analisa VCIRS untuk Sistem Pakar Deteksi Dini Penyakit....	IV-9
4.2.1 Pembangunan Pengetahuan .....	IV-10
4.2.2 Motor Inferensi .....	IV-17
4.3 Pengembangan Perangkat Lunak .....	IV-20
4.3.1 Diagram Konteks .....	IV-20
4.3.2 Diagram Aliran Data ( <i>Data Flow Diagram</i> ) .....	IV-21
4.3.2.1 DFD Level 1 Sistem Pakar Deteksi Dini Penyakit .....	IV-22
4.3.2.2 DFD Level 2 Proses 2 Pengolahan Data Master .....	IV-23
4.3.2.3 DFD Level 2 Proses 3 Konsultasi .....	IV-24
4.3.2.4 DFD Level 3 Proses 2.1 Pengelolaan Data Pengguna ....	IV-25
4.3.2.5 DFD Level 3 Proses 2.2 Pengelolaan Data Rule .....	IV-25
4.3.2.6 DFD Level 3 Proses 2.3 Pengelolaan Data Variable .....	IV-26
4.3.2.7 DFD Level 3 Proses 2.4 Pengelolaan Data Penanganan .....	IV-27
4.3.3 <i>Entity Relational Diagram</i> (ERD) .....	IV-27
4.3.3.1 Kamus Data .....	IV-28
4.3.3.2 Kamus Data <i>Login</i> .....	IV-29
4.3.3.3 Kamus Data Gejala .....	IV-30

4.3.3.4 Kamus Data <i>Node</i> .....	IV-30
4.3.3.5 Kamus Data <i>Rule</i> .....	IV-30
4.3.3.6 Kamus Data Penanganan .....	IV-30
4.3.3.7 Kamus Data <i>User</i> .....	IV-31
4.3.3.8 Kamus Data Gejala <i>User</i> .....	IV-31
4.4 Tahapan Proses Sistem Pakar .....	IV-31
4.5 Antar Muka Pengguna Sistem .....	IV-32
4.5.1 Perancangan Struktur Menu .....	IV-32
4.5.2 Perancangan Antar Muka.....	IV-33
BAB V. IMPLEMENTASI DAN PENGUJIAN.....	V-1
5.1 Implementasi Sistem.....	V-1
5.1.1 Alasan Pemilihan Perangkat Lunak .....	V-1
5.1.2 Batasan Implementasi .....	V-1
5.1.3 Lingkungan Implementasi.....	V-2
5.1.4 Hasil Implementasi.....	V-2
5.1.4.1 Menu Utama/ <i>form Login</i> .....	V-2
5.1.4.2 Menu Form Konsultasi .....	V-3
5.1.4.3 Menu Form Hasil Interview .....	V-5
5.2 Pengujian Sistem .....	V-6
5.2.1 Lingkungan Pengujian Sistem .....	V-6
5.2.2 Hasil Pengujian <i>Black Box</i> .....	V-7
5.2.2.1 Identifikasi Dari Sistem Yang Menggunakan <i>Black Box</i> Pada Butir Pengujian Menu Deteksi Dini Penyakit .....	V-7
5.2.2.2 Identifikasi Butir Pengujian Pertanyaan .....	V-7
5.2.3 Identifikasi Dari Sistem Yang Menggunakan <i>User</i> <i>Acceptance Test</i> .....	V-8
5.2.3.1 Hasil Dari <i>User Acceptance Test</i> .....	V-9
5.2.4 Perbandingan Data Training dengan Data Hasil Pakar .....	V-10
5.4 Kesimpulan Pengujian .....	V-13

BAB VI. PENUTUP .....	VI-1
6.1 Kesimpulan.....	VI-1
6.2 Saran .....	VI-1

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR RIWAYAT HIDUP

## DAFTAR TABEL

Tabel	Halaman
4.1 Gejala Penyakit .....	IV-7
4.2 Jenis Penyakit.....	IV-7
4.3 Penyakit dan Gejala.....	IV-8
4.4 Penyakit dan Solusi .....	IV-9
4.5 Kejadian Dari <i>Variable</i> (gejala) Kasus I Dalam KB.....	IV-11
4.6 Kejadian dari <i>Node</i> Kasus I Dalam KB .....	IV-11
4.7 Analisa Nilai.....	IV-11
4.8 Perbedaan Dari Kasus Lama dan Kasus Baru.....	IV-13
4.9 Kejadian Dari Variabel (gejala) Kasus II Opsi IV Dalam KB.....	IV-14
4.10 Kejadian Dari Node Kasus II Opsi IV Dalam KB .....	IV-15
4.11 Analisa Nilai.....	IV-15
4.12 Analisa Nilai.....	IV-16
4.13 Proses DFD Level 1 Sistem Pakar Deteksi Dini Penyakit.....	IV-22
4.14 Aliran Data DFD Level 1 Sistem Pakar Deteksi Dini Penyakit.....	IV-22
4.15 Proses DFD Level 2 Proses 2 Pengolahan Data Master .....	IV-23
4.16 Aliran Data DFD Level 2 Proses 2 Pengolahan Data Master .....	IV-23
4.17 Proses DFD Level 2 Proses 3 Konsultasi.....	IV-24
4.18 Aliran Data DFD Level 2 Proses 3 Konsultasi .....	IV-24
4.19 Proses DFD Level 3 Proses 2.1 Pengelolaan Data Pengguna .....	IV-25
4.20 Aliran Data DFD Level 3 Proses 2.1 Pengelolaan Data Pengguna...	IV-25
4.21 Proses DFD Level 3 Proses 2.2 Pengelolaan Data <i>Rule</i> .....	IV-26
4.22 Aliran Data DFD Level 3 Proses 2.2 Pengelolaan Data <i>Rule</i> .....	IV-26
4.23 Proses DFD Level 3 Proses 2.3 Pengelolaan Data <i>Variable</i> .....	IV-26
4.24 Aliran Data DFD Level 3 Proses 2.3 Pengelolaan Data <i>Variable</i> ....	IV-26
4.25 Proses DFD Level 3 Proses 2.4 Pengelolaan Data Penanganan .....	IV-27
4.26 Aliran Data DFD Level 3 Proses 2.4 Pengelolaan Data Penanganan	IV-27
4.27 Keterangan Entitas pada ER-Diagram .....	IV-29



4.28	Kamus Data login.....	IV-29
4.29	Kamus Data gejala .....	IV-30
4.30	Kamus Data <i>node</i> .....	IV-30
4.31	Kamus Data <i>rule</i> .....	IV-30
4.32	Kamus Data penanganan.....	IV-30
4.33	Kamus Data <i>user</i> .....	IV-31
4.34	Kamus Data gejala pasien .....	IV-31
5.1	Butir Pengujian Menu Deteksi Penyakit.....	V-7
5.2	Butir Pengujian Pertanyaan.....	V-7
5.3	Jawaban Kasil Pengujian Kuisisioner.....	V-9
5.4	Hasil Pengujian Sistem .....	V-10
5.5	Diagnosa Pakar (Dokter).....	V-11
5.6	Perbandingan Hasil Pengujian Sistem dan Hasil Diagnosa Dokter..	V-12

## DAFTAR RUMUS

Rumus	Halaman
2.1 VUR .....	IV-8
2.2 Weight.....	IV-8
2.3 CD .....	IV-8
2.4 NUR .....	IV-9
2.5 RUR .....	IV-9
2.6 CF Kombinasi .....	II-13

## DAFTAR SINGKATAN

AI	: <i>Artificial Intelligent</i>
CF	: <i>Certainty Factor</i>
DFS	: <i>Depth First Search</i>
KB	: <i>Knowledge Base</i>
NUR	: <i>Node Usage Rate</i>
RB	: <i>Rule Base</i>
RBS	: <i>Rule Based System</i>
RDR	: <i>Riple Down Rule</i>
RUR	: <i>Rule Usage Rate</i>
VCIRS	: <i>Variable Centered Intelligent Rule System</i>
VUR	: <i>Variable Usage Rate</i>

## DAFTAR RUMUS

Rumus	Halaman
2.1 VUR .....	IV-8
2.2 Weight.....	IV-8
2.3 CD .....	IV-8
2.4 NUR .....	IV-9
2.5 RUR .....	IV-9
2.6 CF Kombinasi .....	II-13

## DAFTAR ISTILAH

<i>Algoritma</i>	= Urutan langkah-langkah yang sistematis
<i>Artificial Intelligence</i>	= Kecerdasan buatan
<i>Backward Chaining</i>	= Pencocokan dari kanan ( <i>THEN</i> )
<i>Blackboard</i>	= Area dalam memori yang digunakan untuk menyimpan kejadian sementara
<i>Case</i>	= Kasus
<i>Candidate Variable.</i>	= Kandidat Variabel
<i>Certainty Factor.</i>	= Faktor kepastian
<i>Consultation Environment</i>	= Lingkungan konsultasi
<i>Conclusion Values</i>	= Nilai kesimpulan
<i>Context Diagram</i>	= Diagram konteks
<i>Credit</i>	= Kejadian dari variabel
<i>Data Base Management System</i>	= Perangkat lunak untuk memanipulasi data
<i>Data Base</i>	= Data Base
<i>Data Flow Diagram</i>	= Diagram Aliran Data
<i>Development Environment</i>	= Lingkungan pengembangan
<i>Entity</i>	= Entitas
<i>Explanation Subsystem</i>	= Subsistem penjelasan
<i>Expert System</i>	= Sistem pakar, sistem yang dapat menggantikan seorang ahli
<i>Expert System Builder</i>	= Pembangun Sistem Pakar
<i>Event-Driven Programming</i>	= Pemrograman terkendali kejadian
<i>Forward Chaining</i>	= Pencocokan fakta dari kiri ( <i>IF-THEN</i> )
<i>Flowchart</i>	= Aliran data
<i>Hardware</i>	= Perangkat keras
<i>Inference Engine</i>	= Motor inferensi
<i>Input</i>	= Data yang dimasukkan
<i>Knowledge Base</i>	= Basis Pengetahuan

<b><i>Knowledge Acquisition Subsystem</i></b>	= Subsistem memperoleh pengetahuan
<b><i>Knowledge Engineer</i></b>	= Insinyur pengetahuan
<b><i>Node Usage Rate</i></b>	= Derajat kegunaan Node
<b><i>Relationship</i></b>	= Hubungan
<b><i>Rule Base System</i></b>	= Sistem Berbasis Aturan
<b><i>Rule Usage Rate</i></b>	= Derajat Kegunaan Rule
<b><i>Software</i></b>	= Perangkat lunak
<b><i>User</i></b>	= Pengguna
<b><i>User Interface</i></b>	= Tampilan antar muka pemakai
<b><i>Variable Usage Rate</i></b>	= Derajat kegunaan Variable
<b><i>Weight</i></b>	= Menghitung bobot ( <i>weight</i> ) dari variabel

## DAFTAR LAMPIRAN

Lampiran	Halaman
A      Pembangunan Pengetahuan.....	A-1
B      Rancangan Antar Muka.....	B-1
C      Form Kuisisioner Penelitian Tugas Akhir .....	C-1
D      Daftar Simbol .....	D-1

## BAB V

### IMPLEMENTASI DAN PENGUJIAN

#### 5.1 Implementasi Sistem

Implementasi merupakan tahapan pembuatan sistem yang dilakukan berdasarkan hasil analisa dan perancangan yang telah dilakukan sebelumnya. Pada tahapan implementasi sistem ini diharapkan sistem yang telah dirancang siap dioperasikan pada keadaan yang sebenarnya, sehingga akan diketahui apakah sistem yang dibuat benar-benar dapat menghasilkan tujuan yang diinginkan.

##### 5.1.1 Alasan Pemilihan Perangkat Lunak

Perangkat lunak yang digunakan pada penelitian ini adalah PHP dan MySQL sebagai *databasenya*. Pertimbangan ini didasarkan pada:

1. PHP dapat dipakai hampir disemua web server yang ada.
2. PHP dan MySQL memiliki kecepatan dalam eksekusi perintah, kemampuan menangani jutaan *request* secara bersamaan.
3. MySQL merupakan *software* sistem manajemen *database* yang sangat populer di kalangan pemrograman web dengan menggunakan *database* sebagai sumber pengelolaan data.
4. Akses *database* menggunakan MySQL lebih fleksibel.
5. MySQL merupakan *database* yang terkenal dengan kecepatan dan keamanannya.
6. MySQL dapat menampung data dalam jumlah yang besar.

##### 5.1.2 Batasan Implementasi

Batasan implementasi dari Tugas Akhir ini adalah:

1. Aplikasi menggunakan bahasa pemrograman PHP dan *database* MySQL.
2. Aplikasi berbasis web
3. *User* hanya menjawab pertanyaan yang diajukan sistem



4. Pertanyaan diperoleh dari gejala yang telah *diinputkan* oleh pakar (dokter).
5. Untuk masalah ketidakpastian yang terjadi akan digunakan *Certainty Factor* untuk menyelesaikannya. Dengan cara menghitung nilai kepastian untuk tiap gejala.
6. Untuk keakuratan hasil diagnosa pasien dianjurkan menemui dokter.

### 5.1.3 Lingkungan Implementasi

Lingkungan implementasi merupakan tempat dibangunnya aplikasi dan tempat diujinya aplikasi tersebut. Lingkungan implementasi tersebut berupa perangkat lunak dan perangkat keras yang digunakan, yaitu:

1. Perangkat keras

Perangkat keras yang digunakan memiliki spesifikasi sebagai berikut:

- a. *Processor* : Intel Core 2 Duo
- b. *Memory* : 1014 MB
- c. *Hardisk* : 120 GB

2. Perangkat lunak

- a. Sistem Operasi : Microsoft Windows XP Professional
- b. Bahasa Pemrograman : PHP
- c. DBMS : *Database MySQL*
- d. *Browser* : *Mozilla Firefox*

### 5.1.4 Hasil Implementasi

Hasil implementasi meliputi tampilan menu utama, konsultasi dan hasil interview.

#### 5.1.4.1 Menu Utama/ Form Login

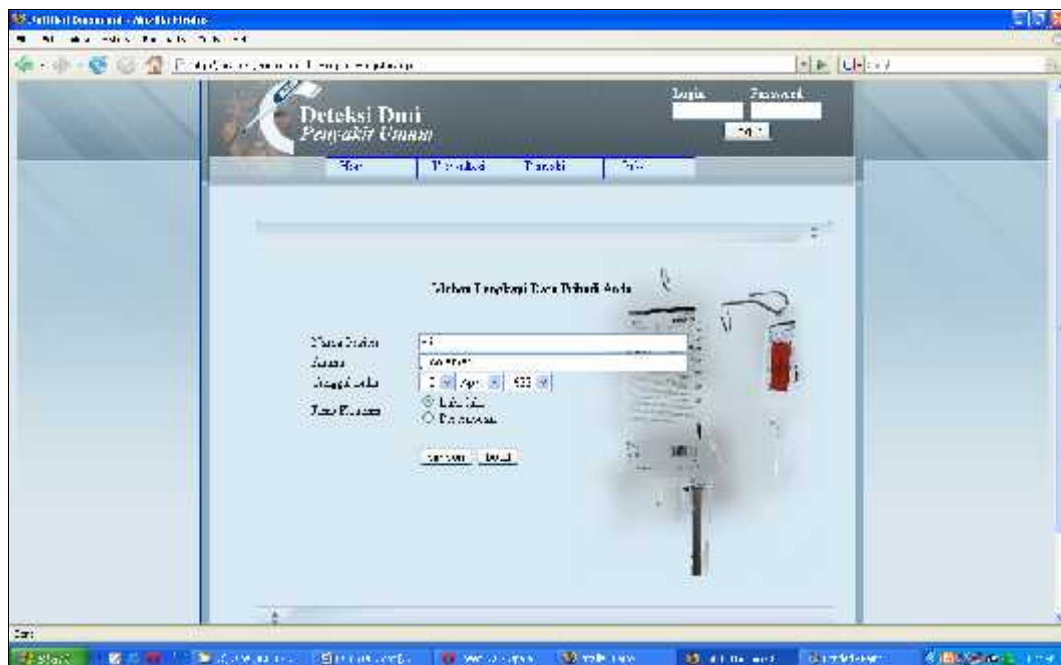
*Form Login* digunakan oleh pakar untuk masuk ke halaman administrator dengan *input username* dan *password* terlebih dahulu. Jika *login* benar pakar dapat masuk ke menu admin. *Form* ini juga dapat digunakan langsung oleh pasien untuk melakukan diagnosa dan melihat info mengenai penyakit umum tersebut. Berikut ini adalah gambar dari *form Login*:



Gambar 5.1 Tampilan *Form Login*

#### 5.1.4.2 Menu Form Konsultasi

Jika *user* ingin melakukan diagnosa, maka *user* dapat mengklik menu konsultasi pada halaman utama. Sebelum memulai konsultasi, sistem meminta *user* mengisi data diri terlebih dahulu. Berikut tampilannya:



Gambar 5.2 Tampilan *Form Identitas Pasien*

Setelah *user input* data diri, *user* dapat melakukan diagnosa dengan cara menjawab pertanyaan-pertanyaan yang diajukan oleh sistem. Berikut tampilan dari menu diagnosa:

Pertanyaan yang muncul pertama sekali diambil dari rule dengan nilai RUR terbesar yaitu demam berdarah#1, kemudian lihat gejala dengan nilai VUR terbesar dari rule tersebut yaitu “gelisah,ujung kaki dan tangan dingin”. Jadi pertanyaan yang muncul pertama kali adalah:

The screenshot shows a web application titled "Deteksi Dini Penyakit Umum". At the top right, there are fields for "Login" and "Password" with a "Login" button below them. Below the header, there is a navigation bar with four tabs: "Home", "Konsultasi", "Pengantar", and "Info". The main content area has a light blue background. In the center, there is a question in Indonesian: "Apakah anda mengalami Gelisah, ujung kaki dan tangan dingin". Below the question are three buttons: "batal" (cancel), "ya" (yes), and "tidak" (no). To the right of the question, there is a small graphic of a white envelope with a red stamp that says "UNIVERSITAS".

Gambar 5.3 Tampilan *Form* Diagnosa Pertanyaan Pertama

*User* menjawab “Ya”, maka akan muncul pertanyaan berikutnya. Tampilannya adalah sebagai berikut.

The screenshot shows the 'Deteksi Dini Penyakit Umum' web application interface. At the top, there is a header with a logo on the left and login fields for 'Login' and 'Password' on the right, with a 'Login' button below them. A navigation bar contains links for 'Home', 'Konsultasi', 'Penyakit', and 'Info'. The main content area features a large text box with the question 'Apakah anda mengalami Pucet'. Below this, there are three buttons: 'Ya', 'Tidak', and 'Tidak Tahu'. To the right of the buttons is a small graphic of a notepad with a pencil.

Gambar 5.4 Tampilan *Form* Diagnosa Pertanyaan Kedua

User menjawab “Ya”, maka akan muncul pertanyaan berikutnya. Tampilannya adalah sebagai berikut.

#### 5.1.4.3 Menu Form Hasil Interview

Menu hasil akhir interview dari deteksi dini penyakit merupakan menu yang menampilkan hasil keseluruhan deteksi yang dijalankan oleh sistem berdasarkan jawaban dari *user*.

The screenshot shows the 'Deteksi Dini Penyakit Umum' web application interface displaying the final interview results. The header and navigation bar are identical to the previous screenshot. The main content area displays 'Interview selesai' in a large text box. Below this, there is a box containing the text 'Kesimpulan malaria (Kryskiman : 97.41 %)' and a small graphic of a notepad with a pencil. At the bottom, there are three buttons: 'Kembali', 'Selesai', and 'Lanjutkan Pemeriksaan'.

Gambar 5.5 Tampilan *Form* Hasil Diagnosa



## 2. Perangkat lunak

Sistem ini akan diuji dengan perangkat lunak menggunakan:

- a. Sistem operasi Windows XP Professional
- b. Bahasa pemrograman PHP
- c. Database My-SQL
- d. Browser Mozilla Firefox

### 5.2.2 Hasil Pengujian *Black box*

Prekondisi : layar untuk pengelolaan data deteksi dini penyakit umum dengan menggunakan metode *variable centered intelligent rule system*.

#### 5.2.2.1 Identifikasi Dari Sistem Yang Menggunakan *Black box* Pada Butir Pengujian Menu Deteksi Dini Penyakit

Tabel 5.1 Butir Pengujian Menu Deteksi Penyakit

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Test
01	Klik menu “Konsultasi”	mengklik menu “Konsultasi”	Klik menu “Konsultasi”	Tampil Layar Pertanyaan	Diterima

#### 5.2.2.2 Identifikasi Butir Pengujian Pertanyaan

Tabel 5.2 Butir Pengujian Pertanyaan

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Hasil Yang Diharapkan	Kriteria Evaluasi Hasil	Hasil Tes
01	Menjawab pertanyaan yang ada dalam sistem	Menjawab Pertanyaan “Ya” bila merasakan dan “Tidak” bila tidak merasakan	Ex: Pertanyaan: 1. apakah anda mengalami gelisah, ujung kaki dan tangan dingin? Ya 2. apakah anda mengalami pucat? Ya	Kesimpulan hasil konsultasi : Demam Berdarah. Hal yang harus dilakukan: 1. beri penderita minum sebanyak-banyaknya	Masukan Sesuai Format	Diterima

			3. apakah anda mengalami muntah darah? Ya 4. apakah anda mengalami nyeri ulu hati ? Ya 5. Apakah anda mengalami perdarahan di hidung? Ya 6. Apakah anda mengalami Panas tinggi? Ya 7. Apakah anda mengalami bercak bintik merah pada kulit? Ya	2.kompres penderita dengan air hangat 3. berikan obat penurun panas.		
--	--	--	--	---	--	--

### 5.2.3 Identifikasi Dari Sistem Yang Menggunakan *User Acceptance Test*

Cara pengujian dengan menggunakan *User Acceptance Test* adalah dengan membuat angket yang didalamnya berisi pertanyaan seputar Tugas Akhir ini, misalnya pertanyaan mengenai pendapat *user* tentang sistem yang dibuat. Angket disertai nama, pekerjaan dan tanda tangan yang mengisi angket. Banyaknya pertanyaan yang ada pada angket sekitar delapan pertanyaan dan berbentuk objektif, dimana yang mengisi angket dapat memilih mana jawaban yang sesuai dengan masalah yang sedang dihadapi. Angket diisi oleh beberapa mahasiswa, pegawai swasta, bidan dan perawat. Banyaknya responden yang mengisi angket adalah sebanyak 10 orang.

### 5.2.3.1 Hasil Dari *User Acceptance Test*

Hasil dari *user acceptance test* dengan cara pengisian kuisioner menjelaskan apakah sistem pakar yang dibangun layak atau tidak dalam menentukan deteksi dini penyakit. Daftar pertanyaan kuisioner yang diajukan dapat dilihat pada lampiran C.

Adapun jawaban dari kuisioner yang telah disebarkan sebagai berikut.

Tabel 5.3 Jawaban hasil pengujian kuisioner

NO	PERTANYAAN	JAWABAN		
		YA	TIDAK	RAGU-RAGU
1	Apakah sebelumnya Bapak/Ibu/Saudara/i pernah menggunakan sistem tertentu yang mengarah kepada deteksi dini penyakit?	-	8	2
2	Apakah sebelumnya Bapak/Ibu/Saudara/i pernah melihat sistem yang sama yaitu Sistem Pakar Untuk Deteksi Dini Penyakit Umum berbasis Web Menggunakan Metode <i>Variable-Centered Intelligent Rule System</i> ?	2	6	2
3	Setelah Bapak/Ibu/Saudara/i mengetahui dan menggunakan aplikasi ini, menurut Bapak/Ibu/Saudara/i sudah baguskah dari segi tampilan atau <i>interface</i> ?	6	2	2
4	Apakah Sistem Pakar untuk Deteksi Dini Penyakit Umum ini perlu diterapkan?	8	-	2
5	Menurut Bapak/Ibu/Saudara/i bagaimana penggunaan navigasi atau menu-menu yang tersedia dari aplikasi ini, apakah ada kesulitan dalam penggunaannya?	4	4	2
6	Dari segi isi, apakah ada informasi yang diberikan oleh Sistem Pakar ini?	10	-	-



7	Pada saat sistem ini dijalankan, apakah ada kesalahan atau <i>error</i> pada salah satu menu yang disediakan?	2	8	-
8	Apakah setelah ada aplikasi ini, Bapak/Ibu/Saudara/i merasa terbantu dalam mendeteksi dini penyakit umum?	10	-	-

Dari hasil pengujian kuisioner yang telah disebarkan, maka dapat diambil kesimpulan tentang sistem pakar untuk deteksi dini penyakit ini dilihat dari 3 komponen dalam kuisioner sebagai berikut:

1. Segi *Interface*

Sistem ini sudah dikatakan layak karena berdasarkan jawaban dari pertanyaan no.3 mengenai segi tampilan/ *interface*, 60% kuisioner menjawab ya.

2. Segi akurasi

Hasil jawaban yang diberikan menyatakan bahwa sistem ini dapat membantu menentukan deteksi dini suatu penyakit karena 100% kuisioner memperoleh informasi dari sistem ini dan terbantu dalam deteksi suatu penyakit.

#### 5.2.4 Perbandingan Data Training Dengan Data Hasil Pakar

Pengujian dilakukan untuk mengukur tingkat validasi antara hasil yang dikeluarkan oleh aplikasi dengan hasil dari pakar (dokter). Dengan tujuan untuk melihat validitas dari sistem.

Tabel 5.4 Hasil Pengujian Sistem

No	Gejala	Demam Berdarah	Tipus	Malaria	Maag
1	Panas Tinggi $\geq 39^{\circ}\text{C}$	Ya	Ya	Ya	-
2	Nyeri Ulu Hati	Ya	-	-	Ya
3	Bercak Bintik Merah di Kulit	Ya	-	-	-
4	Perdarahan di Hidung	Ya	-	-	-
5	Muntah Darah	Ya	-	-	-
6	pucat	Ya	-	-	-
7	Gelisah, Ujung Kaki dan Tangan Dingin	Ya	-	-	-
<b>Hasil analisa</b>		<b>DB 99.89%</b>			

8	Badan lemas	Ya	-	-
9	Tubuh Menggigil	Ya	Ya	-
10	Denyut nadi Lemah $\leq$ 80-100/menit	Ya	-	-
11	Sakit kepala	Ya	-	-
12	Nyeri otot	Ya	Ya	-
13	Kehilangan nafsu makan	Ya	-	-
14	Sakit perut	Ya	-	-
15	Gangguan dalam metabolisme	Ya	-	-
<b>Hasil Analisa</b>		<b>Tipus (99.99%)</b>		
16	Demam berkala (suhu tubuh tidak stabil)	Ya	-	
17	Timbul keringat dingin	Ya	-	
<b>Hasil Analisa</b>			<b>Malaria (97.41%)</b>	
18	Mual			Ya
19	Perut kembung			Ya
20	Nyeri lambung			Ya
21	Kurang nafsu makan			Ya
22	Sakit saat BAB			Ya
23	Muntah $\geq$ 5 kali dala satu hari			Ya
<b>Hasil Analisa</b>			<b>Maag (99.98%)</b>	

Tabel 5.5 Diagnosa Pakar (Dokter)

No	Gejala	Demam Berdarah	Tipus	Malaria	Maag
1	Panas Tinggi $\geq$ 39°C	Ya	Ya	Ya	-
2	Nyeri Ulu Hati	Ya	-	-	Ya
3	Bercak Bintik Merah di Kulit	Ya	-	-	-
4	Perdarahan di Hidung	Ya	-	-	-
5	Muntah Darah	Ya	-	-	-
6	Pucat	Ya	Ya	-	-
7	Gelisah, Ujung kaki dan tangan dingin	Ya	ya	-	-
<b>Hasil analisa</b>		<b>Demam Berdarah (100%)</b>			
8	Badan Lemas		Ya	-	-
9	Tubuh Menggigil		Ya	Ya	-

10	Denyut Nadi Lemah $\leq 80-100/\text{menit}$	Ya	-	-
11	Sakit Kepala	Ya	-	-
12	Nyeri Otot	Ya	Ya	-
13	Kehilangan Nafsu Makan	Ya	-	-
14	Sakit Perut	Ya	-	-
15	Mengalami Gangguan Pencernaan	Ya	-	-
<b>Hasil Analisa</b>		<b>Tipus (100%)</b>		
16	Demam Berkala (Suhu tubuh tidak stabil)		Ya	-
17	Timbul Keringat Dingin		Ya	-
<b>Hasil Analisa</b>			<b>Malaria (100%)</b>	
18	Mual			Ya
19	Perut Kembung			ya
20	Nyeri Lambung			Ya
21	Kurang Nafsu Makan			Ya
22	Sakit Saat BAB			Ya
23	Muntah $\geq 5$ kali dalam satu hari			Ya
<b>Hasil Analisa</b>				<b>Maag (100%)</b>

Berikut tabel perbandingan antara hasil pengujian sistem dan hasil diagnosa dokter:

Tabel 5.6 Perbandingan Hasil Pengujian Sistem dan Hasil diagnosa Dokter

Penyakit	Pengujian Pada Sistem	Diagnosa Dokter	Persentase Hasil Pengujian Sistem (%)	Persentase Diagnosa Dokter (%)
Demam Berdarah	√	√	99,89	100
Tipus	√	√	99,99	100
Malaria	√	√	97,41	100
Maag	√	√	99,98	100

Dari tabel pengujian 5.9 diatas, maka dapat disimpulkan bahwa tidak ditemukannya perbedaan yang signifikan antara persentase hasil yang dikeluarkan oleh aplikasi dengan persentase hasil dari pakar (dokter).

### **5.3 Kesimpulan Pengujian**

Berdasarkan hasil pengujian, aplikasi sistem pakar untuk deteksi dini penyakit menggunakan metode *variable centered intelligent rule system* mampu memberikan hasil yang optimal serta layak digunakan dan juga tidak ditemukannya perbedaan yang signifikan antara persentase hasil yang dikeluarkan oleh aplikasi dengan persentase hasil dari pakar (dokter).

## **BAB VI**

### **P E N U T U P**

#### **6.1 Kesimpulan**

Setelah melalui tahap pengujian pada sistem pakar untuk deteksi dini penyakit umum, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Telah berhasil dibangun sebuah sistem pakar untuk deteksi penyakit menggunakan *Variable Centered Intelligent Rule System* (VCIRS) yang dapat menghasilkan kesimpulan berupa deteksi penyakit dan solusinya.
2. Berdasarkan pengujian user *acceptance test* dan perbandingan hasil pengujian sistem dan hasil diagnosa dokter, aplikasi ini mampu memberikan hasil yang optimal serta layak digunakan dan juga tidak ditemukannya perbedaan yang signifikan antara persentase hasil yang dikeluarkan oleh aplikasi dengan persentase hasil dari pakar (dokter).

#### **6.2 Saran**

Beberapa hal yang dapat diungkap sebagai saran untuk pengembangan Sistem Pakar Deteksi Dini Penyakit adalah sebagai berikut:

1. Merancang sistem berbasis *Wireless Application Protocol* (WAP), sehingga lebih leluasa digunakan dimana saja.
2. Pilihan jawaban pada sistem ini bisa dikembangkan lagi dalam bentuk pilihan ganda, karena untuk menjawab pertanyaan tidak cukup dengan memilih jawaban 'YA' dan 'TIDAK' saja.

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Penggunaan teknologi komputer telah merambah ke berbagai bidang, salah satunya adalah bidang medis. Dalam bidang medis, teknologi komputer ini dimanfaatkan antara lain untuk membantu mendiagnosa jenis-jenis penyakit dan cara untuk mengatasinya berbeda-beda untuk tiap jenis penyakit. Banyak penderita yang ingin tahu secara persis apa penyakit yang dideritanya, penyebabnya dan bagaimana cara pengobatan dan penanganan apabila gangguan kesehatan tersebut terjadi.

Bagi orang awam gejala seperti pusing dan demam biasanya tidak ditanggapi dengan serius. Padahal gejala tersebut bisa juga merupakan petunjuk adanya penyakit yang lebih serius. Misalnya, gejala seperti demam. Demam bisa merupakan gejala awal dari penyakit demam berdarah, malaria atau tipus. Seperti orang awam sering terbentur pada masalah pendiagnosaan gejala penyakit dan cara penanggulangannya secara dini. Kebanyakan orang keberatan untuk pergi ke dokter untuk berobat dengan berbagai alasan bahwa berobat ke dokter mahal, banyak antriannya sehingga terlalu lama menunggu, terlalu jauh tempat dokternya dan alasan lainnya. Karena berbagai alasan tersebut mereka berpendapat bahwa dapat menyembuhkan penyakit tersebut dengan mengetahui penanggulangannya secara dini. Oleh karena itu, penulis ingin membuat suatu sistem pakar yang dapat digunakan sebagai alat bantu untuk membantu penanggulangan secara dini penyakit yang diderita pasien.

Metode yang digunakan untuk memasukkan *cases* (kasus-kasus) yang ada ke dalam *Knowledge Base* (basis pengetahuan) adalah VCIRS (*Variable-Centered Intelligent Rule System*) karena VCIRS memiliki kelebihan dalam *knowledge building* (pembangunan pengetahuan) sekaligus mempunyai kemampuan yang baik dalam proses inferensi. Namun demikian sistem pakar masih memiliki keterbatasan yaitu tidak disediakannya suatu mekanisme untuk menilai tingkat kepercayaan

terhadap sebuah kasus semua kasus dianggap memiliki tingkat kepercayaan 100%, padahal dalam dunia nyata jarang sekali ditemukan tingkat kepercayaan dengan nilai tersebut. Dalam hal ini untuk menyelesaikan masalah digunakan metode Faktor Kepastian (*Certainty Factor*). CF adalah ukuran atau tingkat kepercayaan seseorang terhadap *rule* yang ada. Meliputi tingkat kepercayaan terhadap suatu *variable* dalam suatu *rule* dan tingkat kepercayaan *rule* itu sendiri.

Berdasarkan beberapa hal yang telah dijelaskan diatas, maka untuk penulisan Tugas Akhir (TA) ini penulis akan memberi judul “**Sistem Pakar Deteksi Dini Penyakit Berbasis WEB Menggunakan Metode *Variable-Centered Intelligent Rule System* (VCIRS)**”. Penulisan tugas akhir ini diharapkan dapat diterima dan dipakai juga sebagai pedoman bagi masyarakat luas agar dapat membedakan antara gangguan ringan atau keadaan yang menuntut bantuan dari dokter.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka permasalahan dalam Tugas Akhir ini adalah, “Bagaimana merancang dan membangun aplikasi sistem pakar berbasis web yang digunakan untuk melakukan diagnosa awal penyakit.

## **1.3 Batasan Masalah**

Agar lebih terarah dan tidak menyimpang dari tujuan semula, maka penulis membuat batasan-batasan yaitu:

1. Metode inferensi sistem pakar yang dipakai adalah *Variable-Centered Intelligent Rule System* akan dikombinasikan dengan *certainty factor*.
2. Mesin inferensi yang digunakan sebagai otak dari sistem pakar ini adalah *forward chaining*.
3. Sistem pakar ini hanya mendiagnosa beberapa gejala-gejala utama penyakit yang menyerang manusia seperti demam berdarah, tipus, malaria dan maag.

#### **1.4 Tujuan Penelitian**

Tujuan dari pelaksanaan Tugas Akhir ini adalah merancang dan membangun suatu sistem pakar tentang penanggulangan penyakit secara dini pada penyakit demam berdarah, tipus, malaria dan maag.

#### **1.5 Sistematika Penulisan**

Sistematika penulisan tugas akhir ini dibagi menjadi 6 (enam) bab yang masing-masing bab telah dirancang dengan suatu tujuan tertentu. Berikut penjelasan tentang masing-masing bab:

##### **BAB I     Pendahuluan**

Merupakan deskripsi umum dari tugas akhir ini, yang meliputi: latar belakang masalah, rumusan masalah, batasan masalah, tujuan penyusunan tugas akhir serta sistematika pembahasan tugas akhir.

##### **BAB II    Landasan Teori**

Berisi penjelasan tentang teori dasar yang akan diterapkan dalam tahap analisis sistem, yang meliputi: konsep dasar sistem, sistem pakar (*expert system*), *Variable-Centered Intelligent Rule System* (VCIRS), Pemodelan analisis dan penyakit.

##### **BAB III   Metodologi Penelitian**

Bab ini terdiri atas: tahapan penelitian, pengumpulan data, analisa kebutuhan sistem, perancangan perangkat lunak, implementasi, pengujian dan kesimpulan dan saran.

##### **BAB IV   Analisa dan Perancangan**

Berisi tentang analisa sistem dengan menggunakan VCIRS, analisa VCIRS untuk sistem pakar deteksi dini penyakit, pengembangan perangkat lunak, tahapan proses sistem pakar dan antar muka pengguna sistem.



## BAB V Implementasi dan Pengujian

Pada bab ini berisi implementasi sistem, pengujian sistem dan kesimpulan pengujian.

## BAB VI Penutup

Dalam bab ini berisikan kesimpulan dan saran.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Konsep Dasar Sistem**

*Webster's Dictionary* mendefinisikan “sistem” sebagai “serangkaian atau tatanan hal-hal yang saling berhubungan untuk membentuk suatu kesatuan atau keseluruhan organik, serangkaian kenyataan, prinsip, aturan, dan lain-lain, yang diklasifikasikan dan diatur di dalam bentuk teratur dengan maksud memperlihatkan suatu rencana logis yang menghubungkan bagian-bagian yang berbeda” (Pressman, 2002).

Dari pengertian yang didapat dari *Webster's Dictionary* dapat diambil kesimpulan bahwa sistem berbasis komputer adalah serangkaian atau tatanan elemen-elemen yang diatur untuk mencapai tujuan yang ditentukan sebelumnya melalui pemrosesan informasi (Pressman, 2002). Untuk mencapai tujuan tersebut, sistem berbasis komputer menggunakan berbagai elemen sistem (Pressman, 2002):

1. Perangkat lunak. Program komputer, struktur data dan dokumen yang berhubungan yang berfungsi untuk mempengaruhi metode logis, prosedur dan kontrol yang diperlukan.
2. Perangkat keras. Perangkat elektronik yang memberikan kemampuan penghitungan, dan perangkat elektromekanik yang memberikan fungsi dunia *eksternal*.
3. Manusia. Pemakai dan operator perangkat lunak dan perangkat keras.
4. *Database*. Kumpulan informasi yang besar dan terorganisasi yang diakses melalui perangkat lunak
5. Dokumentasi. Manual, formulir dan informasi deskriptif lainnya yang menggambarkan penggunaan dan atau pengoperasian sistem.
6. Prosedur. Langkah-langkah yang menentukan penggunaan khusus dari masing-masing elemen sistem atau konteks prosedural di mana sistem berada.

## 2.2 Sistem Pakar (*Expert System*)

Sistem pakar dibuat untuk menirukan seorang pakar atau ahli. Sistem pakar adalah paket *hardware* dan *software* yang digunakan sebagai pengambil keputusan dan penyelesaian masalah yang dapat mencapai *level* yang setara bahkan kadang melebihi seorang pakar atau ahli pada satu area masalah yang spesifik dan biasanya lebih sempit (Subakti, 2006). Sistem pakar merupakan cabang dari aplikasi *Artificial Intelligent* (AI).

Kepakaran didapatkan dari seorang pakar dan dipindahkan ke dalam komputer. Lalu kepakaran tersebut disimpan dan *user* dapat meminta saran spesifik yang dibutuhkannya. Komputer dapat mencari, mengolah dan menampilkan kesimpulan yang spesifik. Dan seperti seorang pakar, saran tersebut dapat dimanfaatkan oleh orang yang bukan pakar berikut penjelasannya yang berisi logika dibalik saran tersebut.

Tujuan dibangunnya sistem pakar bukanlah untuk menggantikan pakar, tetapi hanya untuk membuat pengetahuan dan pengalaman para pakar tersebut tersimpan dan tersedia lebih luas dan leluasa.

### 2.2.1 Konsep Dasar Sistem Pakar

Konsep dasar sistem pakar mengandung keahlian, ahli/pakar, pengalihan keahlian, inferensi, aturan dan kemampuan menjelaskan.

Keahlian adalah pengetahuan yang spesifik yang didapatkan dari *training*, membaca, dan pengalaman. Ahli/pakar adalah derajat atau *level* dari keahlian. Pengalihan keahlian yang dimaksud adalah pengalihan keahlian dari para ahli ke komputer untuk kemudian dialihkan lagi ke orang lain yang membutuhkan baik orang awam maupun untuk para pakar sebagai asistennya.

Fitur khas dari sistem pakar adalah kemampuan untuk *reasoning*. Keahlian disimpan dalam *Knowledge Base* dan sistem memiliki akses kedalam *database*, maka komputer diprogram untuk dapat berinferensi. Inferensi ini dilakukan oleh komponen yang disebut dengan *inference engine* (mesin inferensi) yang didalamnya terdapat prosedur-prosedur yang berkaitan dengan penyelesaian masalah.

Fitur unik lain dari sistem pakar adalah kemampuan untuk menjelaskan nasehat atau rekomendasi yang diberikan. Penjelasan ini dilakukan oleh subsistem yang disebut dengan *justifier* atau *explanation subsystem*.

### 2.2.2 Struktur Sistem Pakar

Sistem Pakar terdiri dari 2 bagian pokok, di antaranya (Kusumadewi, 2003):

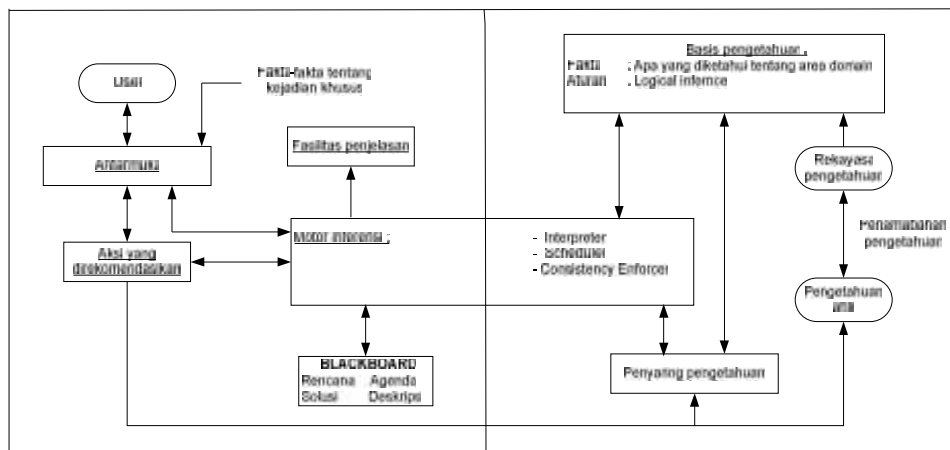
1. **Lingkungan pengembangan** (*development environment*)

Digunakan sebagai pembangunan sistem pakar baik dari segi pembangunan komponen maupun basis pengetahuan.

2. **Lingkungan konsultasi** (*consultation environment*)

Digunakan oleh seorang yang bukan ahli untuk berkonsultasi.

Gambar berikut akan menjelaskan struktur sistem pakar:



Gambar 2.1 Struktur Sistem Pakar (Sumber : Kusumadewi, 2003)

### 2.2.3 Komponen Sistem Pakar

Sistem pakar terbagi menjadi beberapa komponen. Komponen-komponen yang terdapat pada sistem pakar adalah sebagai berikut:

1. *Knowledge Acquisition Subsystem*

Pengetahuan dapat diperoleh dari seorang pakar, teks buku atau laporan penelitian, dengan dukungan dari seorang *knowledge engineer*.

2. *Basis Pengetahuan (Knowledge Base)*

Basis pengetahuan merupakan inti dari suatu sistem pakar, yaitu berupa representasi pengetahuan dari pakar. Basis pengetahuan tersusun atas fakta dan kaidah. Fakta adalah informasi tentang objek, peristiwa, atau situasi.

Kaidah adalah cara untuk membangkitkan suatu fakta baru dari fakta yang sudah diketahui. Basis pengetahuan merupakan representasi dari seorang pakar.

3. Mesin Inferensi (*Inference Engine*)

Mesin inferensi berperan sebagai otak dari sistem pakar. Mesin inferensi berfungsi untuk memandu proses penalaran terhadap suatu kondisi, berdasarkan pada basis pengetahuan yang tersedia. Di dalam mesin inferensi terjadi proses untuk memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi penalaran dan strategi pengendalian. Strategi pengendalian berfungsi sebagai panduan arah dalam melakukan proses penalaran. Terdapat dua teknik pengendalian yang sering digunakan, yaitu *forward chaining* dan *backward chaining*.

4. *Blackboard*

*Blackboard* adalah tempat menyimpan sementara untuk memproses rencana, agenda, solusi dan deskripsi masalah yang didapat dari *Knowledge Base* selama sesi konsultasi.

5. *User*

*User* yang dimaksud dalam sistem pakar ini adalah: (1) klien (bukan pakar) yang menginginkan nasehat. (2) *Learner* (pelajar) untuk mempelajari bagaimana sistem pakar menyelesaikan permasalahan. (3) *Expert System Builder* yang meningkatkan *Knowledge Base*-nya. (4) Pakar, disini sistem pakar berperan sebagai asistennya.

6. Antarmuka Pemakai (*User Interface*)

Fasilitas ini digunakan sebagai perantara komunikasi antara pemakai dengan sistem.

7. *Explanation Subsystem*

Kemampuan penelusuran kebenaran dari konklusi yang didapat dari sumber-sumbernya.

#### 8. *Knowledge Refining System*

Dengan komponen ini pakar mampu untuk menganalisis kinerja dari sistem pakar.

### 2.3 *Variable-Centered Intelligent Rule System (VCIRS)*

VCIRS merupakan teknik persilangan dari *Rule Based-System* (RBS) dan *Ripple Down Rules* (RDR).

#### 1. *Rule Base System* (RBS)

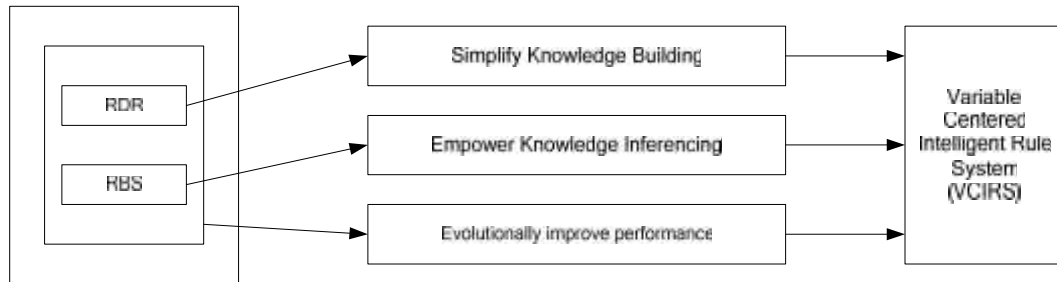
Sistem Berbasis Aturan (SBA – *Rule Base Systems* (RBS)) adalah sistem yang baik untuk memberikan jawaban dari pertanyaan mengenai *What* (apa), *How* (bagaimana) dan *Why* (mengapa) dari *Rule Base* (RB) selama proses inferensi. Jawaban dan penjelasannya disediakan dengan baik. Masalah dengan SBA adalah ia tidak dapat secara mudah menjalankan proses *Knowledge Acquisition* dan ia tidak dapat *update rule* (aturan) secara otomatis. Hanya pakar yang dapat *update Knowledge Base* (KB) secara manual dengan dukungan dari *Knowledge Engineer*.

#### 2. *Ripple Down Rule* (RDR)

RDR diciptakan untuk mengatasi permasalahan utama dari sistem pakar. RDR dapat melakukan akuisisi dengan cepat dan sederhana secara ekstrim tanpa bantuan dari *Knowledge Engineer*. Pengguna tidak perlu menguji *Rule Base* untuk mendefinisikan *rule* baru. Pakar hanya perlu mendefinisikan *rule* baru secara benar mengklasifikasikan contoh yang diberikan, dan sistem dapat menentukan dimana suatu *rule* harus ditempatkan dalam hirarki *rulanya*. Keterbatasan dari RDR adalah kekurangan dalam hal inferensi yang berdayaguna. Tidak seperti SBA yang dilengkapi dengan inferensi melalui *forward chaining* dan *backward chaining*, RDR menggunakan *Depth First Search* (DFS) yang memiliki kekurangan dalam hal fleksibilitas dalam hal penjawaban pertanyaan dan penjelasan yang tumbuh dari inferensi yang berdayaguna.

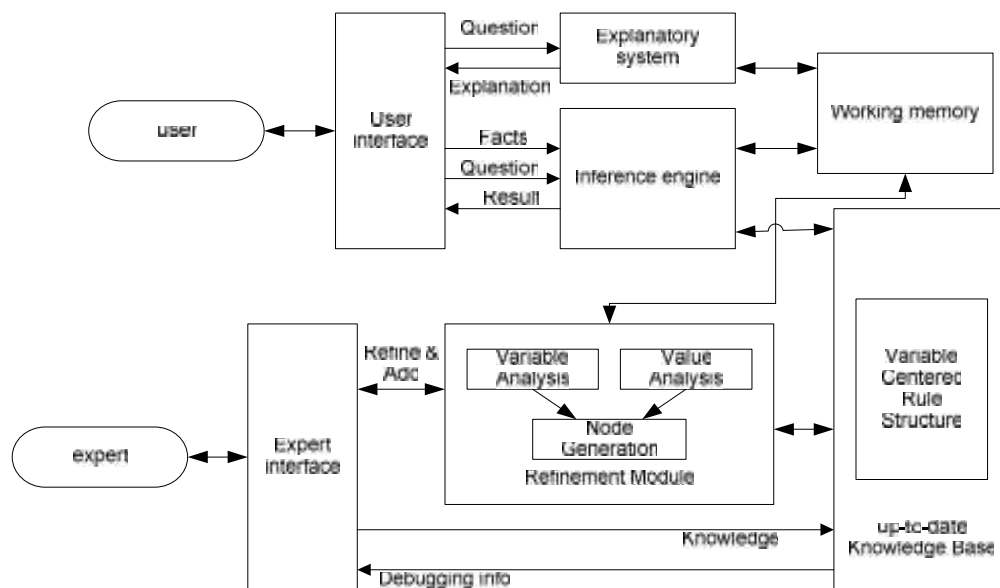
### 2.3.1 Arsitektur Sistem

*Variable-Centered Intelligent Rule System (VCIRS)* merupakan perkawinan dari RBS dan RDR. Arsitektur sistem diadaptasi dari RBS dan ia mengambil keuntungan-keuntungan yang ada dari RDR.



Gambar 2.2 Diagram Metode VCIRS

Sistem ini mengorganisasi *Rule Base* dalam struktur spesial sehingga pembangunan pengetahuan lebih sederhana, inferensia pengetahuan yang berdayaguna dan peningkatan evolusional dari kinerja sistem dapat didapatkan pada waktu yang sama.



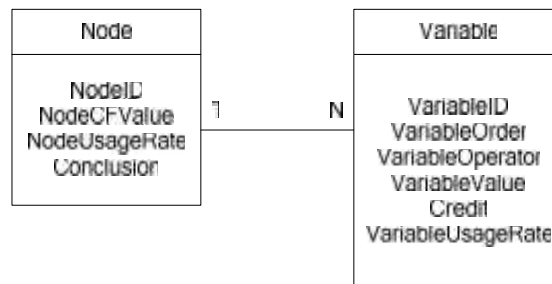
Gambar 2.3 Arsitektur VCIRS

Gambar 2.3 memperlihatkan arsitektur dari VCIRS yang mengadaptasi dari arsitektur RBS. Hanya saja ditambahkan dua buah modul baru yang disebut dengan *Variable-Centered Rule Structure* dan *Refinement Module* ke dalamnya.

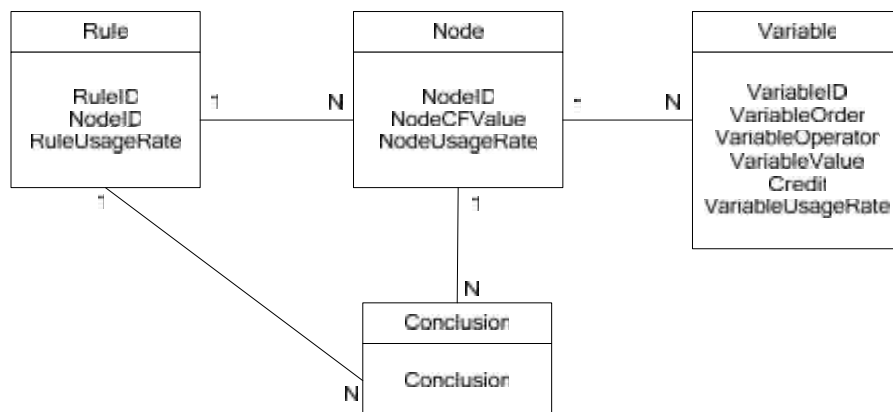
*Variable-Centered Rule Structure* digunakan untuk mengelola KB yang *up-to-date*. *Refinement Modul* mempunyai 3 tugas yaitu: analisis variabel, analisis nilai dan pembangkit *node*.

### 2.3.1.1 *Variable-Centered Rule Structure*

Struktur ini mengandung 2 struktur, yang disebut dengan *Node Structure* (struktur *node*) dan *Rule Structure* (struktur *rule*). Struktur *node* menyimpan kasus yang dipresentasikan oleh pakar dan menghitung kejadian dari setiap kasus. Struktur *rule* menyimpan rangkaian dari *node* yang direpresentasikan oleh struktur *node*. Setiap *node* dalam KB memiliki rangkaian, yaitu paling tidak ada satu *node* untuk satu *rule*.



Gambar 2.4 *Node Structure*



Gambar 2.5 *Rule Structure*



### 2.3.1.2 Refinement Module

Ada 3 tugas dalam *Refinement Module*, tugas-tugas tersebut berguna dalam proses *up-to-date Knowledge Base*. Tugas-tugas tersebut adalah:

1. Analisis variabel. Analisis variabel menentukan manakah variabel/*node* yang paling penting.
2. Analisis nilai. Analisis nilai menentukan seberapa sering sebuah *rule/node/variabel* itu digunakan.
3. Pembangkitan *node*. Pembangkitan *node* adalah hasil dari analisis variabel dan analisis nilai.

Di dalam *Variable-Centered Rule Structure* sistem mengetahui *node* mana yang di-*shared* (sama-sama menggunakan) oleh berbagai *rule*, dan variabel mana yang di-*shared* oleh *node*. Semakin banyak *rule* yang memakai suatu *node*, maka *node* tersebut akan semakin penting. Pertimbangan yang sama terjadi pada variabel yang di-*shared* didalam *node*.

Proses analisis nilai, yang disebut dengan *usage assignment* (pemberian nilai kegunaan), adalah untuk menentukan derajat kegunaan (*Usage Degree*) dari *rule/node/variabel* dalam KB. *Usage assignment* menggunakan informasi yang disimpan dalam *Variable-Centered Rule Structure*. Ada 3 jenis *Usage Degree*, yaitu:

1. *Variable Usage Rate* (VUR) digunakan untuk mengukur tingkat kegunaan dari suatu variabel di dalam *node* yang sedang dan telah digunakan.

$$VUR_i = Credit_i \times Weight_i \quad (2.1)$$

$$Weight_i = NS_i \times CD_i \quad (2.2)$$

$$CD_i = \frac{VO_i}{TV} \quad (2.3)$$

2. *Node Usage Rate* (NUR) untuk mengukur tingkat kegunaan suatu *node* pada pengekseskuan (*firing*).

$$NUR_j = \frac{\sum_{i=1}^N VUR_{ij}}{N}, VUR_{ij} \text{ untuk variabel ke-} i \text{ dalam } node \ j \quad (2.4)$$

3. *Rule Usage Rate* (RUR) yang mengukur tingkat kegunaan suatu *rule* pada pengekseskuan (*firing*).

$$RUR_k = \frac{\sum_{j=1}^N NUR_{jk}}{N}, NUR_{jk} \text{ untuk } node \text{ ke-} j \text{ dalam } rule \ k \quad (2.5)$$

Keterangan:

$Credit_i$  : Kejadian dari variabel  $i$  dalam *Node Structure*

$Weight_i$  : Menghitung bobot (*weight*) dari variabel ke *node* yang memilikinya

$NS_i$  : Jumlah *node* yang berbagi (*sharing*) variabel  $i$

$CD_i$  : Derajat Kedekatan dari variabel  $i$  dalam *node*  $j$ .

$VO_i$  : Urutan dari variabel  $i$  dalam suatu *node*

$TV$  : Total variabel yang dimiliki oleh suatu *node*

Makin besar indikator kegunaan, maka makin bergunalah nilai tersebut dan begitu pula sebaliknya.

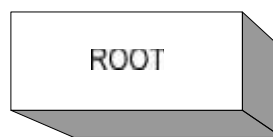
Pembangkit *node* bekerja berdasarkan hasil dari analisis variabel dan nilai. Informasi mengenai *shared node*/variabel dari analisis variabel berguna untuk memilih kandidat yang baik untuk membuat kombinasi. *Shared node*/variabel yang paling tinggi nilainya berarti bahwa ialah yang merupakan *node*/variabel terpenting dalam KB yang ada, karena ia digunakan di banyak tempat pada struktur saat ini. *Usage degree* dengan nilai tertinggi yang diperoleh dari analisis nilai berarti bahwa *node*/variabel tersebut memiliki kejadian (*occurrence*) tertinggi di dalam struktur.

### 2.3.2 Pembangunan Pengetahuan

Pembangunan pengetahuan dalam VCIRS mengizinkan pakar untuk membangun KB dari tidak ada sama sekali (*scratch*) atau untuk *update* KB yang telah ada.

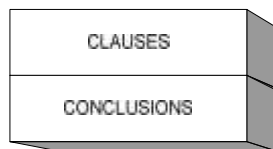
#### 2.3.2.1 Pohon Inferensi VCIRS

Jika dalam VCIRS kita masih belum memiliki apa-apa dalam KB-nya. Maka pohon inferensi dapat digambarkan sebagai berikut (Subakti, 2006).

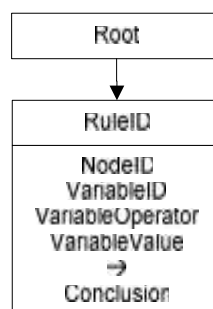


Gambar 2.6 *Knowledge Base* dalam VCIRS saat masih kosong

Setiap 1 *node* dalam KB adalah merupakan representasi dari 1 *rule* dalam RBS (Subakti, 2006).



Gambar 2.7 Representasi 1 *node* dalam KB pada VCIRS



Gambar 2.8 Arsitektur Pohon Inferensi VCIRS

Setiap kasus baru yang diolah dalam VCIRS dibandingkan dengan beberapa *rule* yang telah ada dalam KB. Lalu diberikan suatu daftar perbedaan yang memuat 2 bagian, yaitu: bagian (A) untuk kasus lama dan bagian (B) untuk kasus baru (Subakti, 2006).

Berdasarkan isi dari suatu kasus, VCIRS akan mencari *node* yang layak. Adapun algoritma untuk mencari *node* yang layak adalah sebagai berikut:

1. Prioritas utama; yaitu menemukan *node* sempurna, yaitu *node* yang mengandung *variableIDs* (memiliki operator yang sama) dan nilai konklusi (*conclusion value*) yang sama.
2. Prioritas kedua; temukan *node* yang mengandung *variableIDs* yang sama (tanpa memiliki operator yang sama) yang juga memiliki nilai-nilai konklusi (*conclusion values*) yang sama.
3. *Node* yang memiliki paling tidak satu *variableID* yang sama yang juga memiliki paling tidak satu konklusi yang sama.
4. *Node* yang memiliki paling tidak satu *VariableID* yang sama.

Sedangkan algoritma untuk membangun basis pengetahuan adalah sebagai berikut:

1. Jika *node* yang layak ditemukan, maka sistem akan memberikan tabel perbedaan yang memuat 2 bagian, yaitu bagian yang berisi kondisi *rule* lama (A) dan bagian yang berisi kondisi *rule* baru (B). Kemudian sistem akan memberikan beberapa opsi berdasarkan tabel perbedaan tersebut. Adapun opsi tersebut adalah:
  - a. Pakar menerima *rule* yang telah ada, tak memasalahkan adanya *case* baru yang tersedia.
  - b. *Rule* baru yang akan dihasilkan adalah negasi dari kondisi pada bagian A yang berbeda pada daftar perbedaan.
  - c. *Rule* baru yang akan dihasilkan adalah kondisi pada bagian B yang berbeda pada daftar perbedaan.
  - d. *Rule* baru yang akan dihasilkan adalah gabungan negasi bagian A yang berbeda dan bagian B yang berbeda.

- e. *Rule* baru yang akan dihasilkan semuanya berasal dari *case* baru yang tersedia tanpa melihat daftar perbedaan, dan letaknya langsung di bawah *root* imajiner. Dengan kata lain ini adalah *rule* yang terletak pada *level* puncak KB.
2. Jika *node* yang layak tidak ditemukan maka sistem akan membuat *node* baru pada *level* puncak KB (*parentnya* root).

Jika suatu kasus diklasifikasikan tidak benar/salah maka hal yang harus dilakukan adalah menghentikan *rule* tersebut sehingga tidak digunakan dalam proses inferensi kemudian membangun kasus baru yang benar pada level puncak KB. Untuk menghentikan suatu *rule* cukup ditambahkan suatu tanda padanya yang menandakan bahwa *rule* ini tidak boleh dipakai lagi (artinya konklusinya tidak boleh digunakan, namun *clausenya* masih dapat digunakan oleh *rule* dibawahnya). Hal tersebut disebut dengan *stopping rule*.

### 2.3.3 Faktor Kepastian (*Certainty Factor*)

*Certainty Factor* digunakan untuk memecahkan masalah utama pada sistem pakar. *Certainty Factor* (CF) adalah ukuran/tingkat kepercayaan seseorang terhadap *rule* yang ada. Ada 2 macam faktor kepastian yang digunakan yaitu; faktor kepastian yang diisikan oleh pakar bersama dengan aturan dan faktor kepastian yang diberikan oleh pengguna. Faktor kepastian yang diisikan oleh pakar menggambarkan kepercayaan pakar terhadap hubungan antara *antecedent* dan konsekuen pada aturan kaidah produksi. Faktor kepastian yang diisikan oleh pengguna menggambarkan kepercayaan pengguna terhadap suatu *antecedent*. Untuk mengetahui faktor kepastian oleh pengguna tidak mudah, karena sulit bagi pengguna untuk memperkirakan besarnya nilai kepastian terhadap element *antecedent* sesuai dengan standar yang diberikan oleh pakar.

$$CF_{kombinasi} = \begin{cases} CF_1 + CF_2(1 - CF_1) & \text{kedua} - \text{duanya} > 0 \\ \frac{CF_1 + CF_2}{1 - \min([CF_1], [CF_2])} & \\ CF_1 + CF_2(1 - CF_1) & \text{kedua} - \text{duanya} < 0 \end{cases} \quad (2.6)$$

#### 2.3.4 Proses Inferensia

Proses inferensia dapat dilakukan dengan beberapa pendekatan yaitu *backward chaining* dan *forward chaining*.

1. *Backward Chaining* (penalaran mundur)

Penalaran dimulai dari tujuan atau hipotesa, baru dicocokkan dengan keadaan awal atau fakta-fakta yang ada.

2. *Forward Chaining* (penalaran maju)

Pelacakan dimulai dari keadaan awal (informasi atau fakta yang ada) dan kemudian dicoba untuk mencocokkan dengan tujuan yang diharapkan.

Selama proses inferensia, VCIRS memperlakukan sebuah *rule* sebagai rangkaian dari *node* (*rule* dalam RBS). Ia mengabaikan isi konklusi dari setiap *node*, kecuali konklusi pada *node* terakhir sebagai konklusi dari *rule*. Dari titik pandang ini inferensia memperlakukan sebuah *rule* sebagai *rule* (besar) dimana *clause part*-nya mengandung semua *clause part* dalam setiap *node* dari rangkaian, dan *conclusion part*-nya adalah konklusi dari *node* terakhir. Sehingga dari sini, operator *clause* adalah operator *AND* (dari semua *clauses*), yang juga merupakan jenis operator konklusi jika ada lebih dari satu nilai konklusi dalam suatu *node* (dan lalu ia menjadi *node* terakhir dalam suatu *rule*) (Subakti, 2006)

#### 2.3.5 Evaluasi Sistem

Seperti halnya RDR, VCIRS hanya memberi perhatian pada variabel (*clause part*) daripada bagian konklusi. Konklusi tidak mempunyai peranan dalam sistem; ia hanyalah hasil yang begitu saja diperoleh. Ini berarti kita dapat mengabaikan konklusi yang terang-terang tanpa resiko apa pun.

Ketidakpentingan konklusi selama proses penelusuran *tree rule* memiliki konsekuensi: tak ada mekanisme untuk melakukan *forward* dan *backward chaining* seperti dalam RBS. RDR tidak menyebutkan mengenai inferensia, sebab RDR ditujukan utamanya untuk akuisisi pengetahuan yang cepat dan sederhana, bukan untuk inferensia. Inferensia dalam RDR dilakukan pada waktu yang sama dengan waktu pengguna melakukan pembangunan pengetahuan dengan penyediaan *case* dan mengikuti sistem bekerja. Dia dapat memilih untuk hanya menelusuri *tree rule* selama operasi tanpa membuat *rule* baru, yang artinya dia hanya ingin melakukan inferensia (yaitu *forward chaining* sederhana). Fakta bahwa dia menelusuri *tree* (inferensia) dan mengupdate KB pada waktu yang sama mengandung arti bahwa verifikasi dari KB dilakukan pada saat berjalannya sistem. VCIRS mewarisi kedua keuntungan ini, yaitu akuisisi pengetahuan yang mudah dan sederhana dan verifikasi sambil jalan (*verification-on-the-fly*).

Sebagai tambahan untuk pendekatan penelusuran *tree* seperti RDR untuk membuat pembangunan (*update*) pengetahuan mudah, VCIRS juga menyediakan mekanisme untuk melakukan transformasi KB sehingga pengguna dapat melakukan inferensia RBS yang berdayaguna. Yang dibutuhkan oleh VCIRS adalah pengguna harus konsisten dalam menggunakan *VariableID* baik pada *clause* maupun *conclusion part*, dan ia mampu untuk mengelola konsistensi *rule-rule* logik dalam keseluruhan KB.

Dibandingkan dengan RDR, VCIRS memiliki dayaguna yang sama dalam pembangunan pengetahuan. Ia juga menyimpan ruang untuk struktur *rule* yang tidak menyimpan setiap rangkaian *nodenya*. Ia cukup mengingat posisi dari setiap *node* dan selanjutnya adalah mudah untuk menyusun ulang *rule* sebagai rangkaian dari beberapa *node*. VCIRS menyimpan setiap kejadian dari *case* dalam *Node Structure* untuk setiap *node*, membantu sistem menghasilkan *rule* baru dari perbaikan pengetahuan, dimana RDR tak dapat melakukan hal ini.

Dibandingkan dengan RBS, VCIRS dapat melakukan *verification-on-the-fly* pengetahuan, dimana RBS tidak dapat melakukannya. Dalam RBS, verifikasi harus dilakukan oleh pakar secara manual yang amat memakan waktu dan mempunyai kecenderungan untuk terjadinya inkonsistensi yang lebih buruk.

Dalam RBS, *rule* yang inkonsistensi cenderung akan berdampak pada *rule* yang lain. Hal ini bukan merupakan hal yang serius dalam VCIRS, karena *rule* yang inkonsisten hanya berakibat pada *sharing nodes (rules)* yang dimiliki oleh beberapa *rule* saja, bukan keseluruhan *rule*. Walaupun beberapa *nodes (rules)* memiliki nilai-nilai yang sama dengan yang lain, kecuali *node* dalam rangkaian yang sama, maka tak ada apa pun yang terjadi.

VCIRS selalu melakukan kalkulasi ulang untuk VUR pada variabel yang terkait, jika suatu variabel dimasukkan kedalam *Variable-Centered Rule Structure*. Ia juga mengkalkulasi ulang NUR dari semua *node* yang menggunakan variabel tersebut, dan meng-*update* RUR dari semua *rule* yang menggunakan *node* ini. hal tersebut akan terlihat sangat membosankan, namun ia berguna untuk memandu pengguna dalam proses pembangunan dan inferensia pengetahuan. Ini juga berguna untuk mendukung pembangkitan *node*.

Disaat melakukan pembangkitan *node*, VCIRS meminta pengguna untuk mengkonfirmasi *rule/node* yang sedang dibangkitkan, sebab pembetulan/pembenaran logik (semantik) dari *rule/node* terbangkitkan masih memerlukan pertimbangan pengguna. Sistem hanya menghasilkan rangkaian *alternatif* dari variabel-variabel yang penting untuk membentuk *node* baru dan rangkaian dari *node* untuk membentuk *rule* masing-masing menurut urutan *relatif* dari variabel dan *node*.

Walaupun VCIRS mengatasi masalah inferensia dari RDR, *algoritma Rete* yang dikembangkan oleh Forgy masih memiliki kinerja yang lebih baik dibandingkan dengan langkah-langkah (*cycles*) yang harus dilakukan sebelum mendapatkan hasil. *Algoritma Rete* digunakan dalam kebanyakan RBS kinerja tinggi, sebab ia adalah metode yang sangat efisien untuk permasalahan pengenalan pola. Kelemahannya adalah ia memiliki kompleksitas ruang tinggi (*high space complexity*). VCIRS yang memiliki *Variable-Centered Rule Structure* menyimpan posisi dari setiap variabel dalam suatu *node* dan *rule* menyediakan kinerja yang baik dan masuk akal, tidak terlalu sempurna seperti halnya *algoritma Rete*, namun masalah kompleksitas ruang cenderung dapat lebih rendah diwujudkan.



## 2.4 Pemodelan Analisis

Model analisis merupakan serangkaian model yang merepresentasikan teknis yang pertama dari sistem. Model analisis harus dapat mencapai tiga sasaran utama, yaitu; (1) untuk menggambarkan apa yang dibutuhkan oleh pelanggan, (2) untuk membangun dasar bagi pembuatan desain perangkat lunak dan (3) untuk membatasi serangkaian persyaratan yang dapat divalidasi begitu perangkat lunak dibangun (Pressman, 2002).

Pada inti model ada kamus data – penyimpanan yang berisi deskripsi dari semua objek data yang dikonsumsi atau diproduksi oleh perangkat lunak. *Entity-relationship diagram* (ERD) menggambarkan hubungan antara objek data. Atribut dari masing-masing objek data yang ditulis pada ERD dapat digambarkan dengan menggunakan deskripsi objek data. *Data Flow Diagram* (DFD) adalah sebuah teknik grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari *input* menjadi *output*. DFD melayani 2 tujuan: (1) untuk memberikan indikasi mengenai bagaimana data ditransformasi pada saat bergerak melalui sistem dan (2) untuk menggambarkan fungsi-fungsi yang mentransformasi aliran data (Pressman, 2002).

## 2.5 Penyakit

Pengertian sakit (*illness*) berkaitan dengan gangguan psiko-sosial yang dirasakan manusia, sedangkan penyakit (*disease*) berkaitan dengan gangguan yang terjadi pada organ manusia. Sakit belum tentu karena penyakit, akan tetapi selalu mempunyai relevansi psiko-sosial (Rosenstock, 1974). Dengan kata lain penyakit adalah istilah medis yang digambarkan sebagai gangguan dalam fungsi tubuh yang menghasilkan berkurangnya kapasitas.

Ada 3 kriteria untuk menentukan apakah seseorang itu sakit atau tidak, antara lain; (Bauman, 1965)

1. Adanya gejala: Naiknya temperatur, nyeri
2. Persepsi tentang bagaimana mereka merasakan: baik, buruk, sakit
3. Kemampuan untuk melaksanakan aktivitas sehari-hari: bekerja, sekolah.

Orang merasakan gangguan dan melakukan tindakan terhadap gangguan, tergantung orang tersebut ingin sakit atau tidak, ingin mencari serta ikut pengobatan atau tidak, serta pikiran, perasaan dan keyakinan terhadap sakit, pengobatan dan profesi kesehatan. Semua pengalaman ini mempengaruhi pengambilan keputusan berobat dan perilaku sehat di masa depan. (Supardi, 1996)

### **2.5.1 Penyakit Demam Berdarah Dengue (DBD)**

Penyakit DBD berkaitan dengan kondisi lingkungan dan perilaku masyarakat. Masyarakat yang kurang peduli kebersihan lingkungan dan ancaman penyakit berbahaya merupakan lokasi yang sangat baik sebagai endemik DBD. Diperlukan kesadaran dan peran aktif semua lapisan masyarakat untuk memberantas DBD dari lingkungan sekitar tempat tinggalnya.

#### **2.5.1.1 Pengertian Penyakit Demam Berdarah Dengue (DBD)**

Penyakit demam berdarah dengue atau yang disingkat sebagai DBD adalah suatu penyakit yang disebabkan oleh virus dengue yang dibawa oleh nyamuk *aedes aegypti* betina lewat air liur gigitan saat menghisap darah manusia.

Selama nyamuk *aedes aegypti* tidak terkontraminasi virus dengue maka gigitan nyamuk demam berdarah tersebut tidak berbahaya. Jika nyamuk tersebut menghisap darah penderita DBD maka nyamuk menjadi berbahaya karena bisa menularkan virus dengue yang mematikan. Untuk itu perlu pengendalian nyamuk jenis *aedes aegypti* agar virus dengue tidak menular dari orang yang satu ke orang yang lain.

#### **2.5.1.2 Gejala Penyakit Demam Berdarah**

Gejala demam berdarah ini ditunjukkan melalui munculnya demam tinggi terus menerus, disertai adanya tanda perdarahan. Selain itu tanda dan gejala lainnya adalah sakit perut, rasa mual, trombositopenia, hemokonsentrasi, sakit kepala, sakit pada sendi dan sakit pada otot.

### **2.5.2 Penyakit Tipus**

Penyakit Tipus berkaitan dengan kondisi lingkungan dan perilaku masyarakat. Masyarakat yang kurang peduli kebersihan lingkungan dan ancaman penyakit berbahaya merupakan lokasi yang sangat baik sebagai sumber penyakit tipus.

#### **2.5.2.1 Pengertian Penyakit Tipus**

Penyakit tipus adalah suatu penyakit yang disebabkan oleh virus yang dibawa oleh tikus lewat air liur sisa makanan maupun kotoran tikus.

Jika seseorang terkena penyakit tipus, maka ia akan merasa lemas karena kurangnya sel darah merah dalam tubuh yang menyebabkan pasien menjadi lemah dan kurangnya metabolisme tubuh.

#### **2.5.2.2 Gejala Penyakit Tipus**

Dalam minggu pertama, keluhan dan gejala menyerupai penyakit infeksi akut pada umumnya, seperti mual, sakit kepala, muntah, nafsu makan menurun, sakit perut, diare pada anak-anak atau sulit buang air besar pada orang dewasa dan suhu tubuh meningkat terutama pada sore dan malam hari.

Setelah minggu kedua gejala menjadi jelas, yaitu demam yang tinggi terus menerus, kulit kering, bibir kering, lidah ditutupi selaput putih yang kotor dan perut kembung.

#### **2.5.3 Penyakit malaria**

Penyakit malaria adalah penyakit infeksi yang disebabkan oleh protozoa parasit yang merupakan golongan plasmodium, dimana proses penularannya melalui gigitan nyamuk anopheles.

##### **2.5.3.1 Gejala Penyakit Malaria**

Masa tunas/inkubasi penyakit ini dapat beberapa hari sampai beberapa bulan yang kemudian barulah muncul tanda dan gejala yang dikeluarkan oleh penderita seperti demam, menggigil, linu atau nyeri persendian, kadang sampai muntah, tampak pucat/anemis, hati serta limpa membesar, air kencing tampak keruh/pekat karena mengandung Hemoglobin, terasa geli pada kulit dan mengalami kekejangan.

Namun demikian, tanda yang klasik ditampakkan adalah adanya perasaan tiba-tiba kedinginan yang diikuti dengan kekakuan dan kemudian munculnya demam dan banyak berkeringat setelah 4 sampai 6 jam kemudian, hal ini berlangsung tiap dua hari. Diantara masa tersebut, mungkin penderita merasa sehat seperti sedia kala. Pada usia anak-anak serangan malaria dapat menimbulkan

gejala aneh, misalnya menunjukkan gerakan/postur tubuh yang abnormal sebagai akibat tekanan rongga otak. Bahkan lebih serius lagi dapat menyebabkan kerusakan otak.

#### **2.5.4 Penyakit maag**

Maag atau radang lambung adalah gejala penyakit yang menyerang lambung dikarenakan terjadi luka atau peradangan pada lambung yang menyebabkan sakit, mulas dan perih pada perut.

##### **2.5.4.1 Gejala Penyakit Maag**

Sakit maag yang fungsional disebabkan oleh beberapa faktor. Keluhan yang timbul bisa terjadi karena akibat peningkatan asam lambung yang berlebihan di dalam lambung. Boleh disebutkan terjadinya banjir asam lambung. Keadaan ini dapat disebabkan oleh makanan dan minuman tertentu, misal makan makanan yang terlalu asam, pedas, minum kopi atau alkohol. Sering terlambat makan dan makan tidak teratur juga dapat menjadi penyebab timbulnya penyakit maag fungsional. Selain itu stres fisik dan psikis juga dapat merangsang produksi asam lambung berlebih sehingga mengakibatkan gangguan maag.

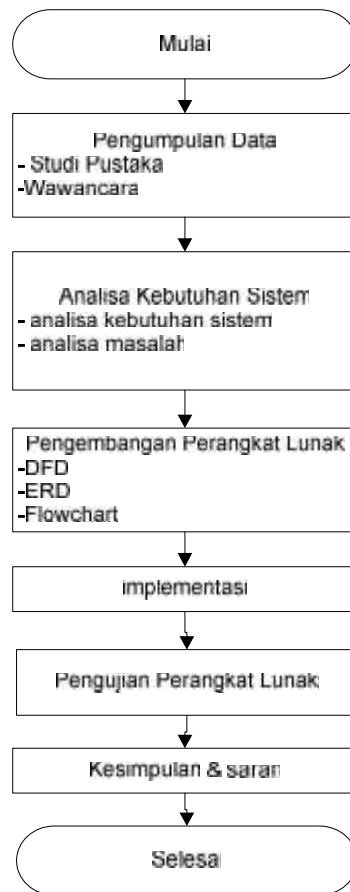
Selain asam lambung yang berlebih, lambung juga bisa terganggu akibat naiknya cairan empedu dari usus dua belas jari menuju lambung., bahkan dapat naik sampai kerongkongan. Adanya cairan empedu yang naik dapat menyebabkan gangguan pada lambung sampai menyebabkan kerusakan dinding dalam lambung.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Tahapan Penelitian

Metodologi penelitian digunakan sebagai pedoman dalam pelaksanaan penelitian agar hasil yang dicapai tidak menyimpang dari tujuan yang telah dilakukan sebelumnya. Metodologi penelitian yang digunakan dalam penyusunan Tugas Akhir ini akan melalui beberapa tahapan yang membentuk sebuah alur yang sistematis. Tahap-tahap yang akan dilalui digambarkan dengan *flowchart* berikut ini:



Gambar 3.1 *Flowchart* Tahapan Penelitian

### **3.2 Pengumpulan Data**

Pengumpulan data merupakan tahap awal yang dilakukan dalam membangun Tugas Akhir ini. Proses pengumpulan data dilakukan dengan 2 (dua) cara, yaitu:

#### **1 Studi Pustaka**

Studi pustaka bertujuan untuk mendapatkan dasar-dasar pengetahuan dan informasi dalam tahap persiapan penelitian dalam membangun Tugas Akhir ini. Melalui studi pustaka diketahui metode yang lebih baik untuk menyelesaikan permasalahan yang terjadi yaitu dengan cara mempelajari buku-buku, artikel-artikel dan jurnal-jurnal yang berhubungan dengan permasalahan penyusunan Tugas Akhir. Beberapa buku yang digunakan sebagai bahan referensi dalam menyelesaikan Tugas akhir ini dapat dilihat pada daftar pustaka.

#### **2 Wawancara**

Setelah melakukan tahapan penelitian awal yaitu studi pustaka, tahapan yang dilakukan selanjutnya adalah wawancara. Wawancara merupakan tahapan pengumpulan data dengan cara berkomunikasi secara langsung dengan pakar dari permasalahan yang diangkat.

Wawancara ini dilakukan untuk mendapatkan data-data yang berhubungan dengan penelitian Tugas Akhir ini. Dalam hal ini wawancara dilakukan kepada pakar (Dr. Rafles Muchlis).

### **3.3 Analisa Kebutuhan Sistem**

Setelah mendapatkan dasar-dasar pengetahuan tahapan berikutnya adalah menganalisa kebutuhan sistem. Sistem ini dimulai dengan menganalisa data yaitu: data data gejala, data penyakit dan data solusi. Data tersebut kemudian disimpan dalam *database* dengan menggunakan metode *Variable-Centered Intelligent Rule System* (VCIRS) dan nantinya akan digunakan dalam proses inferensia menggunakan mekanisme inferensia *Rule Base System* (RBS) dengan metode *forward chaining*.

Dari data-data yang telah di-*input*-kan, proses diagnosa akan dilakukan setelah *user* menjawab pertanyaan-pertanyaan yang diberikan oleh sistem. Langkah-langkah yang terjadi adalah sebagai berikut; sistem memberikan beberapa pertanyaan seputar gejala-gejala dari suatu penyakit kepada pasien, pasien akan memberikan jawaban berdasarkan gejala yang dirasakan.

### **3.3.1 Analisa Masalah**

Masalah yang terjadi pada saat ini adalah lambatnya proses bagi pasien untuk mendapatkan pengobatan secara langsung bila terkena sakit. Pasien harus terlebih dahulu ke dokter/rumah sakit dan mendapatkan pemeriksaan. Pada saat yang bersamaan pasien harus mendapatkan pengobatan secepatnya.

Permasalahan yang terjadi adalah kebanyakan orang malas untuk pergi ke dokter untuk berobat dengan berbagai alasan bahwa berobat ke dokter mahal, banyak antriannya sehingga terlalu lama menunggu, terlalu jauh tempat dokternya dan sebagainya. Banyaknya waktu yang terbuang bagi pasien karena seharusnya sudah mendapatkan pengobatan.

### **3.3.2 Analisa kebutuhan**

Analisa kebutuhan adalah suatu metode menganalisa kebutuhan data yang diperlukan dalam pembuatan sistem ini, yang dibutuhkan adalah:

- a. Data Gejala Penyakit
- b. Data Penyakit
- c. Data Penanggulangan Penyakit

### **3.3.3 Alat Bantu**

Alat bantu adalah alat bantu/*tool* yang dibutuhkan dalam pengolahan data dan pengimplementasian data kedalam sistem. Alat bantu dalam pembuatan sistem ini adalah:

- a. Pemrograman PHP
- b. *Database* MySQL
- c. Appserv
- d. Macromedia Dreamweaver

### **3.4 Perancangan Perangkat Lunak**

Tahap perancangan sistem merupakan tahapan dalam membuat rincian sistem hasil dari analisis menjadi suatu bentuk perancangan agar dipahami oleh pengguna (*user*).

Perancangan sistem meliputi merancang kebutuhan data dengan cara pembuatan diagram konteks, DFD, ERD, *Flowchart*, basis pengetahuan dan antar muka (*user interface*).

### **3.5 Implementasi**

Implementasi sistem merupakan tahap sistem siap dioperasikan pada keadaan yang sebenarnya, sehingga akan diketahui apakah sistem yang dibuat telah menghasilkan tujuan yang diinginkan.

Pada tahap implementasi ini perangkat lunak yang digunakan yaitu PHP dengan *Database* MySQL sehingga memudahkan pengguna untuk menggunakannya.

Mekanisme inferensi yang dipakai dalam penelitian ini adalah *Rule Base System* dengan metode *forward chaining*. Pada tahap ini juga diterapkan *Certainty Factor* untuk menangani ketidakpastian pada basis pengetahuan untuk menunjukkan nilai kepastian dan kebenaran dari hasil diagnosa.

### **3.6 Pengujian**

Pengujian sistem dilakukan pada lingkungan perangkat keras dan lingkungan perangkat lunak. Tahapan dalam pengujian pada sistem ini menggunakan tiga cara yaitu menggunakan *Black Box*, *user acceptance test* dan perbandingan data training dengan data hasil pakar.

### **3.7 Kesimpulan dan Saran**

Kesimpulan merupakan hasil akhir yang didapatkan dari pembahasan sesuai dengan proses-proses yang telah dilakukan sebelumnya. Sedangkan saran merupakan keinginan-keinginan penulis atas kekurangan yang terdapat pada permasalahan yang diangkat sehingga kekurangan tersebut dapat diselesaikan pada pengembangan berikutnya.



## DAFTAR PUSTAKA

- Arhami, Muhammad. "*Konsep Dasar Sistem Pakar*". Yogyakarta : Andi Yogyakarta, Halaman 67, 68, 81. 2005.
- Desiani, Anita. "*Konsep Kecerdasan Buatan*". Palembang : Andi Yogyakarta, Halaman 40. 2005.
- Kristanto, Andi. "*Kecerdasan Buatan*". Klaten : Graha Ilmu, 2003. Halaman 12.
- Kusrini. "*Aplikasi Sistem Pakar, Menentukan faktor kepastian pengguna dengan metode kuantifikasi pertanyaan*". Yogyakarta : Andi Yogyakarta, Halaman 15, 17, 20 - 25. 2008.
- Luger, George F & Stubblefield, Willian A. "*Artificial Intelegence (Structures and Strategies for Complex Problem Solving)*", Addison-Wesley, USA, 1998.
- Pressman, S. Roger, "*Rekayasa Perangkat Lunak*". Yogyakarta : Andi Yogyakarta, Halaman 276, 355 – 360. 2002
- Rosenstock IM. *The health belief and preventive health behavior. Health education monograph*, 1974; 2(4): 354.
- Subakti, Irvan. "*Intelegensia Mesin*". Edisi Jurusan T. Informatika-ITS, Halaman 181 – 217. 2006.
- Subakti, Irvan. "*Thesis Variable-Centered Intelligent Rule System*". National Taiwan University of Science and Technology. Halaman 91- 114. 2006.
- Supardi, sudibyo.
- <http://www.kalbe.co.id/files/cdk/files/17PerilakuSakit111.pdf/17PerilakuSakit111.html>
- Wiston, Patrick Henry. "*Artificial Intelegence Third Edition*", Addison-Wesley Publishing Company, USA, 1993.